

EXHIBIT N

FILED UNDER SEAL

1 IN THE UNITED STATES DISTRICT COURT
2 FOR THE NORTHERN DISTRICT OF CALIFORNIA

3
4
5 GOOGLE LLC,

6 Plaintiff,

7 vs.

No. 3:20-cv-06754-WHA

8 SONOS, INC.,

9 Defendant.

_____/

10
11
12 -- ATTORNEYS' EYES ONLY SOURCE CODE --

13
14 VIDEO-RECORDED DEPOSITION OF KEN MACKAY
15 IN HIS INDIVIDUAL CAPACITY AND AS A
16 FEDERAL RULE 30(B)(6) WITNESS FOR GOOGLE LLC
17 Remote Zoom Proceedings
18 Sunnyvale, California
19 Wednesday, January 25, 2023
20
21
22

23 REPORTED BY:

24 LESLIE ROCKWOOD ROSAS, RPR, CSR 3462

25 Pages 1 - 88

Job No. 5682406

Page 1

| | |
|--|--|
| <p>1 else in the source code?</p> <p>2 A. Yes.</p> <p>3 Q. Now -- okay. So -- so a couple -- going back to</p> <p>4 those two lines that we looked at, that get run, 20 --</p> <p>5 2077 and 2078. 09:39:01</p> <p>6 A. Yes.</p> <p>7 Q. In -- and what I want to ask is I want to -- I</p> <p>8 want to kind of compare this version of the</p> <p>9 multizone_manager file that we're looking at right now to</p> <p>10 the previous versions of this file that may have been 09:39:19</p> <p>11 used to build prior versions of firmware.</p> <p>12 In -- did those prior versions of firmware make</p> <p>13 a call to StopCurrentApp() -- excuse me. Let me ask that</p> <p>14 again.</p> <p>15 Did the source code for the prior versions of 09:39:43</p> <p>16 firmware make a call to StopCurrentApp() as part of the</p> <p>17 RefreshDeviceGroups() function?</p> <p>18 MR. KAPLAN: Object to form.</p> <p>19 THE WITNESS: So prior to me making the change</p> <p>20 that added that call, it didn't include that call. 09:39:57</p> <p>21 Q. BY MR. SHEA: Okay. And then did prior --</p> <p>22 source code for prior versions of firmware make a call to</p> <p>23 AddGroup() as part of the RefreshDeviceGroups() function?</p> <p>24 A. Yes.</p> <p>25 MR. KAPLAN: Object to form. 09:40:23</p> <p style="text-align: right;">Page 26</p> | <p>1 portion of this file. And while you're doing that, I'm</p> <p>2 going to just for the record do two things. First, I'm</p> <p>3 going to introduce this as Exhibit 1321.</p> <p>4 (Exhibit 1321, Application_Manager_IMPL.cc, was</p> <p>5 marked for identification by counsel 09:42:26</p> <p>6 electronically.)</p> <p>7 MR. SHEA: And for the record, I'm also going to</p> <p>8 state that the header of this file is the word</p> <p>9 Application_Manager_IMPL.cc.</p> <p>10 Q. And if you see here, Mr. MacKay, this file, it 09:43:00</p> <p>11 starts here. This printed page starts at line 2043.</p> <p>12 Do you see that?</p> <p>13 A. Yes.</p> <p>14 Q. And is this the StopCurrentApp() function that</p> <p>15 we were just -- that was just referenced in the previous 09:43:14</p> <p>16 file we were looking at?</p> <p>17 A. I assume so. I don't know. I'm not very</p> <p>18 familiar with the application manager source code.</p> <p>19 Q. Okay. So you didn't make any changes or do</p> <p>20 anything to this particular piece of the code? 09:43:39</p> <p>21 A. Not recently.</p> <p>22 Q. Okay. Not as part of this change where you</p> <p>23 added the StopCurrentApp() function call to</p> <p>24 RefreshDeviceGroups()?</p> <p>25 A. That's right. 09:43:56</p> <p style="text-align: right;">Page 28</p> |
| <p>1 Q. BY MR. SHEA: Sorry, Mr. MacKay, was that a</p> <p>2 "yes"?</p> <p>3 A. Yes.</p> <p>4 Q. And in this recent update to the source code</p> <p>5 that -- that you made, Mr. MacKay, did you make any 09:40:37</p> <p>6 changes to the AddGroup() function itself?</p> <p>7 A. No.</p> <p>8 Q. So I want to dive a little deeper on -- on what</p> <p>9 is actually happening with the StopCurrentApp() function</p> <p>10 to make sure I understand it. And I think you had 09:41:00</p> <p>11 mentioned that it calls to the application manager.</p> <p>12 Do I have that right?</p> <p>13 A. Yes.</p> <p>14 Q. I think we have that as well, but actually,</p> <p>15 before we get there, can I have you flip to line 2783. 09:41:18</p> <p>16 A. Yes.</p> <p>17 Q. So do I understand correctly that what's</p> <p>18 happening at 2783 through 2787 is that this is the</p> <p>19 StopCurrentApp() function that we just looked at in the</p> <p>20 prior code, and then that function calls out here to this 09:41:49</p> <p>21 StopCurrentApp() function from the app manager?</p> <p>22 A. Yes.</p> <p>23 Q. Okay, great.</p> <p>24 So then let's -- so if we flip -- I'm going to</p> <p>25 have you go to Bates page 1598. It's a single-page 09:42:04</p> <p style="text-align: right;">Page 27</p> | <p>1 Q. So without -- what I really just want to make</p> <p>2 sure I understand is: What is this actually doing? So</p> <p>3 could you explain for me when we say that a</p> <p>4 StopCurrentApp() function is called on the device, what</p> <p>5 does that StopCurrentApp() function do to the device when 09:44:16</p> <p>6 called?</p> <p>7 A. So my understanding is that any app that's</p> <p>8 currently running will be stopped, or an app is -- well,</p> <p>9 I don't know how to explain an app. I guess it's like a</p> <p>10 piece of code that can be running that's performing 09:44:43</p> <p>11 something for the user.</p> <p>12 Q. Okay. So -- yeah, and that's part of what I</p> <p>13 want to make sure I understand, is when we use the phrase</p> <p>14 "app" in this context, what -- what -- what is an "app"</p> <p>15 here? 09:45:05</p> <p>16 MR. KAPLAN: Object to form.</p> <p>17 THE WITNESS: I just -- I don't know if I have a</p> <p>18 good definition for an app. I could provide examples of</p> <p>19 apps, maybe.</p> <p>20 Q. BY MR. SHEA: I think that would be really 09:45:22</p> <p>21 helpful.</p> <p>22 So -- so would you be able to give me a couple</p> <p>23 examples of what an app would be in this context?</p> <p>24 MR. KAPLAN: Object to form.</p> <p>25 THE WITNESS: So, for example, if you cast 09:45:37</p> <p style="text-align: right;">Page 29</p> |

| | |
|--|---|
| <p>1 Q. BY MR. SHEA: I see. So -- so at least, 2 depending on how the app developer develops that, in some 3 cases, it may be possible that after playback ends, the 4 app will continue to run on the -- the receiver app will 5 continue to run on the player. 09:53:02 6 A. It is possible. So that the cast software 7 allows it. 8 Q. Okay. And -- all right. 9 So I think we've talked a fair amount about what 10 an app is. Now kind of going back to the 09:53:17 11 StopCurrentApp() function, so when that gets run, what 12 will -- what then happens to these kinds of apps that 13 we've been talking about? 14 MR. KAPLAN: Object to form. 15 THE WITNESS: They get stopped. 09:53:41 16 Q. BY MR. SHEA: When you say "they get stopped," I 17 want to make sure we're -- we're clear on what that 18 means. 19 What does it mean to stop an app in this 20 context? 09:53:55 21 A. The -- the code that executes as part of the app 22 stops executing. I guess the app structures are removed 23 from memory, probably. I'm not sure on that one, but I 24 think in general -- in general, it's unloaded. 25 Q. Okay. Is it -- are there circumstances where 09:54:24 Page 34</p> | <p>1 that the file that which the snippet -- or sorry, 2 application_manager_IMPL.cc, I guess it would be. 3 THE REPORTER: Excuse me. Excuse me. 4 Application manager what? 5 THE WITNESS: Oh, sorry. 09:55:58 6 Application_manager_IMPL.cc. 7 Q. BY MR. SHEA: So then is it your understanding 8 that the end result of a StopCurrentApp() function is 9 that the player will no longer be running any receiver 10 app -- any current receiver app at least? 09:56:24 11 A. Yes, that's the intention. 12 Q. But based what you just said, it's possible that 13 it may still be running an app that's preloaded in the 14 background? 15 MR. KAPLAN: Object to form. 09:56:43 16 THE WITNESS: So the background apps are not in 17 a running state. 18 Q. BY MR. SHEA: Okay. Can you explain for me how 19 you distinguish between, you know, an app that's 20 preloaded versus an app that's running? 09:56:56 21 A. So, again, I'm not an expert on the application 22 management, but my understanding is that when we preload 23 an app, we load the code. So we -- typically, they're 24 JavaScript apps, and so we would -- we would load the 25 JavaScript page and perform all of the JavaScript loading 09:57:24 Page 36</p> |
| <p>1 that app would just get -- you talked a little bit 2 earlier about that there's different designations for 3 apps. Some might be considered a current app versus 4 maybe a background app. 5 Is it -- are there circumstances where the 09:54:40 6 StopCurrentApp() function may move the app into a 7 different state like that? 8 MR. KAPLAN: Object to form. 9 THE WITNESS: I'm not sure. I'd have to check 10 the source code. But I think if the app is running, we 09:54:55 11 always stop it completely, and then if we want an 12 instance to be running preloaded in the background, then 13 we would reinstantiate it in that state. 14 Q. BY MR. SHEA: And do you know, are there 15 circumstances where that might happen, where it gets 09:55:15 16 stopped but then automatically preloaded in the 17 background? 18 MR. KAPLAN: Object to form. 19 THE WITNESS: I think it might, but I'm not 20 sure. 09:55:30 21 Q. BY MR. SHEA: If we were trying to get a 22 better -- get better guidance on -- on that piece of the 23 code, where -- where would we want to look; do you know? 24 MR. KAPLAN: Object to form. 25 THE WITNESS: Probably application_manager.cc so 09:55:44 Page 35</p> | <p>1 functions that Chrome performs, I guess. But we wouldn't 2 actually let it start executing. 3 Q. So with respect to this StopCurrentApp() 4 function, will that perform any checking of group state 5 as part of stopping the app? 09:58:01 6 A. No, I don't think so. 7 Q. Is there even any information about a group that 8 would be passed into the StopCurrentApp() function? 9 MR. KAPLAN: Object to form. 10 THE WITNESS: Well, if we look at the source 09:58:26 11 code, it doesn't take any arguments. 12 Q. BY MR. SHEA: Right. And so it's probably a 13 more precise way to -- to kind of say what I was asking, 14 which is that because that source code does not take any 15 arguments, that function, that therein -- thereby means 09:58:43 16 that it -- there's no group information of any kind 17 passed into that function; is that right? 18 A. Correct. There's no -- no group information is 19 passed into the function. That's right. 20 Q. And to your knowledge, would this call to 09:59:01 21 StopCurrentApp() that is -- you added to the 22 RefreshDeviceGroups() function cause a speaker group to 23 be launched? 24 A. Sorry, is the question whether StopCurrentApp() 25 causes the speaker group to be launched? 09:59:25 Page 37</p> |

| | |
|---|--|
| <p>1 that would cause them, for example, to enter the -- the 2 launched state for the group, or I think there's other 3 control signals that they can also receive. 4 Q. BY MR. SHEA: Yeah. I guess I was asking more 5 from a user perspective that -- that if these -- the 10:06:44 6 group is not launched in an unlaunched state, is there 7 anything that the user could do in order to perform some 8 group-level control while that group is in an unlaunched 9 state? 10 MR. KAPLAN: Object to form. 10:07:07 11 THE WITNESS: They could change the group name. 12 Q. BY MR. SHEA: I see. 13 So they could make some changes to the group 14 configuration of that unlaunched group? 15 A. Yes. 10:07:26 16 Q. You've answered a lot of these already. So 17 turning to that AddGroup() function that we talked a bit 18 about, which I think is on -- the call to it in 19 RefreshDeviceGroups() is in 2078, line 2078. 20 Will the AddGroup() function cause a speaker 10:08:02 21 group to be launched? 22 A. No, not directly. 23 Q. And when you say "not directly," is there some 24 indirect pathway you have in your mind? 25 MR. KAPLAN: Object to form. 10:08:26 Page 42</p> | <p>1 message for a newly created static group, it's the case 2 that the AddGroup() function will not result in the 3 static group becoming launched; is that correct? 4 MR. KAPLAN: Object to form. 5 THE WITNESS: Yes. The execution of the 10:11:06 6 AddGroup() function itself will not cause the device to 7 be launched as part of the group. 8 Q. BY MR. SHEA: And is it also true that the 9 AddGroup() function, in this scenario where it's being 10 run as part of the creation of a new static group, will 10:11:24 11 also not indirectly cause that group to be launched 12 through some other pathway of code? 13 A. Well, it does in the sense that it adds the 14 group to the set of groups that are eligible for leader 15 election, and then later, when we call whatever it was, 10:11:50 16 update group leaders, that will cause the connection to 17 be made to the leader, and then the device will receive 18 the launch message if the group is in the launch state. 19 Q. Yeah. And what I'm saying is in that scenario 20 where the group is not in a launch state because it's a 10:12:12 21 newly created group -- 22 A. Oh, okay. 23 Q. -- then in that case, right, it's not going to 24 cause the group to be launched at that point in time; 25 correct? 10:12:25 Page 44</p> |
| <p>1 THE WITNESS: So farther down in the 2 RefreshDeviceGroups() function, on line 2128, there's a 3 call to update group leaders, and that will run leader 4 election. 5 And if a valid leader is known to the device, 10:08:58 6 then at that point the device will connect to the group 7 leader. 8 And once that connection is complete, then if 9 the group is in a launched state, the device that was 10 just added to the group will receive the launch command 10:09:18 11 and start playing as part of the group. 12 Q. In a scenario where this code is being carried 13 out while handling a join group message for a new speaker 14 group that's being created for the first time, do I 15 understand correctly that that speaker group will always 10:09:42 16 start in an unlaunched state? 17 A. If it is a statically defined group, then yes. 18 Q. And by "statically defined group," that also 19 what is referred to in the Google Home app as a speaker 20 group; is that right? 10:10:15 21 A. Yes. 22 MR. KAPLAN: Object to scope. 23 Q. BY MR. SHEA: So maybe tie it together. 24 So in a scenario where the RefreshDeviceGroups() 25 function is being carried out while handling a join group 10:10:35 Page 43</p> | <p>1 MR. KAPLAN: Object to form. 2 THE WITNESS: Yes. Correct. 3 MR. KAPLAN: Rory, we've been going about an 4 hour. Do you want to take a break? 5 MR. SHEA: Yeah, let's do it. 10:12:33 6 THE VIDEOGRAPHER: We are going off the record. 7 The time is 10:12. 8 (Recess.) 9 THE VIDEOGRAPHER: We are back on the record. 10 The time is 10:24. 10:24:01 11 Q. BY MR. SHEA: So, Mr. MacKay, can I have you 12 take a look back at that multizone_manager file again. 13 And I want to ask you just about a couple other parts of 14 it quickly. 15 A. Okay. 10:24:17 16 Q. The first one is a function at line 2766. 17 A. Okay. 18 Q. So as you know, we've been talking most of the 19 morning about a function called StopCurrentApp(). I see 20 here that there's -- at 266, there's a separate function 10:24:35 21 defined called StopApp(). 22 Do you see that? 23 A. Do you mean at line 2766? 24 Q. Thank you. Yeah. 25 A. Yes. 10:24:46 Page 45</p> |

| | |
|--|---|
| <p>1 let's just take one of the players as an example. As I 2 said, the scenario here is that both players were not 3 playing at the time the group was created so they are 4 functioning similarly.</p> <p>5 When the player runs -- when the player makes a 10:40:04 6 call to the StopCurrentApp() function, and the AddGroup() 7 function in this scenario, what will happen to the 8 behavior of the player?</p> <p>9 MR. KAPLAN: Object to form.</p> <p>10 THE WITNESS: So specifically when you call 10:40:38 11 StopCurrentApp(), since there's no current app running, 12 there's -- there's nothing to stop. And then when 13 AddGroup() is called, it will create the multizone group 14 structure for the group.</p> <p>15 And then farther down in RefreshDeviceGroups(), 10:40:59 16 it will start the leader election process for that 17 device.</p> <p>18 And I think -- I forget, but somewhere in that 19 process, we also update the MD&S record for the device to 20 indicate that's part of the group. So it becomes 10:41:21 21 discoverable.</p> <p>22 Q. BY MR. SHEA: Okay. So that process runs, and 23 let me take it kind of piece-by-piece.</p> <p>24 Is -- with respect to the behavior of the player 25 itself, I think you said that because it wasn't engaging 10:41:42 Page 54</p> | <p>1 laid out where we're creating a new group that is a 2 static group, do I understand correctly that the 3 functionality -- the end result of this process will be 4 the creation of a new speaker group that is in an 5 unlaunched state? 10:43:40</p> <p>6 A. Yes.</p> <p>7 Q. And am I correct that at no point during the 8 process that the player runs in order to add itself to 9 the group or memorialize it has been added to the group 10 is that speaker group ever launched even in some 10:44:05 11 temporary capacity?</p> <p>12 A. Yes.</p> <p>13 Q. So then what I want to ask now is a little bit 14 of a variant of the scenario. We've been talking in 15 terms of the player not running a current app. 10:44:31</p> <p>16 What I want to ask you now about is the same 17 scenario except the player is running a current app, but 18 is not engaging in active playback, which we -- I think 19 we talked about earlier is a possibility, that you could 20 be running a current app but not actively engaging in 10:44:56 21 playback at that time.</p> <p>22 Does that make sense?</p> <p>23 MR. KAPLAN: Object to form.</p> <p>24 THE WITNESS: So prior to the user creating the 25 new group, the player A, for example, is -- is running 10:45:10 Page 56</p> |
| <p>1 in playback before it received the join group message, it 2 would remain in the same state after it receives that 3 message; is that right?</p> <p>4 MR. KAPLAN: Object to form.</p> <p>5 THE WITNESS: So if there is no current app to 10:42:00 6 stop, then that's -- it will -- there will still be no 7 current app after StopCurrentApp() runs.</p> <p>8 Q. BY MR. SHEA: Right.</p> <p>9 So in this scenario, then, do I understand 10 correctly that the player will effectively behave in the 10:42:16 11 same way that it would have behaved in prior firmware 12 versions of -- of the software?</p> <p>13 A. Prior to adding the StopCurrentApp() call?</p> <p>14 Q. Yeah.</p> <p>15 MR. KAPLAN: Object to form. 10:42:38</p> <p>16 THE WITNESS: Well, it will still call 17 StopCurrentApp(), but it -- other than that, it won't do 18 anything differently.</p> <p>19 Q. BY MR. SHEA: Right. It's not going to -- the 20 StopCurrentApp() function in this case is not going to 10:42:56 21 cause any change to -- to the operational behavior of the 22 player; correct?</p> <p>23 MR. KAPLAN: Object to form.</p> <p>24 THE WITNESS: Yes.</p> <p>25 Q. BY MR. SHEA: And then in this scenario we've 10:43:12 Page 55</p> | <p>1 some app, but it's not playing media.</p> <p>2 Q. BY MR. SHEA: Exactly.</p> <p>3 A. Yes.</p> <p>4 Q. So in that scenario, can you explain to me what 5 happens when -- or maybe I can just ask you: Is there 10:45:31 6 any change in what happens when the RefreshDeviceGroups() 7 function gets called relative to the scenario we just 8 discussed?</p> <p>9 A. Yes.</p> <p>10 Q. And what is that change? 10:45:45</p> <p>11 MR. KAPLAN: Object to form.</p> <p>12 THE WITNESS: So since there is a current app in 13 this case, in this scenario, the StopCurrentApp() call 14 would cause that app to stop.</p> <p>15 Q. BY MR. SHEA: Now in that scenario, do you agree 10:46:05 16 that the playback state of the player does not change 17 before and after it has been added to the group?</p> <p>18 MR. KAPLAN: Object to form.</p> <p>19 THE WITNESS: So since it wasn't playing media 20 before, stopping the app doesn't cause it to change 10:46:32 21 whether or not it's playing media.</p> <p>22 Q. BY MR. SHEA: And from a -- so from a user's 23 perspective, would -- would this scenario where it's not 24 playing media beforehand, although it is due to -- it is 25 running an app beforehand, will the -- does the media. 10:46:57 Page 57</p> |